FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Multimodal Modular Neural Network Frameworks for VR Interaction: a Comparative Study

Rúben Lourinha Monteiro



Mestrado de Engenharia Informática e Computação

Supervisor: Prof. Rui Pedro Amaral Rodrigues

Second Supervisor: Prof. Ricardo Pereira de Magalhães Cruz

July 25, 2025

Multimodal Modular Neural Network Frameworks for VR Interaction: a Comparative Study

Rúben Lourinha Monteiro

Mestrado de Engenharia Informática e Computação

Approved in oral examination by the committee:

President: Prof. Daniel Filipe Martins Tavares Mendes
Referee: Prof. Filipe Manuel Carvalho Rodrigues Bravo Gonçalves
Referee: Prof. Rui Pedro Amaral Rodrigues

July 25, 2025

Resumo

A Realidade Virtual (RV) oferece experiências imersivas, mas o desenvolvimento de uma interação natural e intuitiva com o utilizador permanece um desafio fundamental. Os sistemas multimodais, que combinam dados de entrada (inputs) como voz, gesto e olhar, constituem uma base promissora para uma solução. A aplicação de Inteligência Artificial (IA) para fundir estas modalidades desbloqueou recentemente um novo e significativo potencial, no entanto, as estruturas(*frameworks*) existentes são, muitas vezes, ou demasiado rígidas ou não estão desenhadas para RV. Esta dissertação aborda esta lacuna, explorando a conceção e a viabilidade de uma *framework* multimodal, inteligente e modular para interações em RV.

Foi concebida uma nova framework de fusão híbrida para ser modular, capaz de processar combinações arbitrárias de modalidades, convertendo-as em representações de características (*features*) padronizadas para serem interpretadas por uma rede neuronal central. Para avaliar a sua eficácia, esta abordagem modular foi comparada (*benchmarked*) com um *pipeline* de fusão tardia não-modular, especificamente adaptado para uma tarefa de RV que envolve inputs visuais, de olhar e de voz. Foi gerado um dataset personalizado num ambiente *Unity* para treinar e testar ambos os modelos em comandos de manipulação de objetos.

Os resultados da análise comparativa revelaram um claro compromisso entre modularidade e desempenho. A *pipeline* especializada de fusão tardia alcançou uma precisão quase perfeita (100%), enquanto que a *framework* de fusão híbrida flexível atingiu uma precisão inferior (89%). O modelo de fusão tardia demonstrou também uma resiliência superior a dados de entrada incompletos. O estudo conclui que, embora uma *framework* modular orientada por IA para interação em RV seja viável, a abstração exigida pela flexibilidade implica, atualmente, um custo significativo em termos de precisão e robustez, tornando os modelos especializados e específicos para cada tarefa a solução mais prática para aplicações de RV que exijam alta precisão.

Palavras-chave: Interação Multimodal, Redes Neuronais, Realidade Virtual (RV), Modularidade, Fala, Olhar, Visão

Abstract

Virtual Reality (VR) offers immersive experiences, but developing natural and intuitive user interaction remains a key challenge. Multimodal systems, which combine inputs like speech, gesture, and gaze, provide a promising foundation for a solution; applying Artificial Intelligence (AI) to fuse these modalities has recently unlocked significant new potential, yet existing frameworks are often either too rigid or not designed for VR. This dissertation addresses this gap by exploring the design and feasibility of an intelligent, modular multimodal framework for VR interactions.

A novel hybrid fusion framework was designed to be modular, capable of processing arbitrary combinations of modalities by converting them into standardized feature representations for a central neural network to interpret. To evaluate its effectiveness, this modular approach was benchmarked against a non-modular, late fusion pipeline specifically tailored to a VR task involving visual, gaze, and speech inputs. A custom dataset was generated in a Unity environment to train and test both models on object manipulation commands.

Results from the comparative analysis revealed a clear trade-off between modularity and performance. The specialized late fusion pipeline achieved near-perfect precision (100%), whereas the flexible hybrid fusion pipeline performed at a lower precision ($\tilde{8}9\%$). The late fusion model also demonstrated superior resilience to incomplete input data. The study concludes that while a modular AI-driven framework for VR interaction is feasible, the abstraction required for flexibility currently comes at a significant cost to accuracy and robustness, making specialized, task-specific models a more practical solution for high-precision VR applications.

Keywords: Multimodal Interaction, Neural Networks, Virtual Reality (VR), Modularity, Speech, Gaze, Vision

Acknowledgements

I´d like to thank my parents, for the constant support, my sister, for the company, my friends, for keeping me sane, and my supervisors and teachers, for keeping me on track and guiding me through this journey.

Rúben Lourinha Monteiro

"Behold, the fruits of empire."

Eioghorain, Brennan Lee Mulligan

Contents

1 Introduction **1**
- 1.1 Context and Motivation . . . 1
- 1.2 Problem . . . 1
- 1.3 Research Statement . . . 2
- 1.4 Research Questions . . . 2
- 1.5 Objectives . . . 2
- 1.6 Document Structure . . . 3

2 State of the Art **4**
- 2.1 Aggregation and Analysis of Various Modalities . . . 4
  - 2.1.1 Filtering of Found Modalities . . . 6
  - 2.1.2 Introduction to Common Modalities . . . 6
  - 2.1.3 Characterization of Candidate Modalities . . . 7
- 2.2 Multimodal Fusion . . . 8
  - 2.2.1 Combining Modalities into Multimodalities . . . 8
  - 2.2.2 Introduction to common multimodal combinations . . . 8
  - 2.2.3 Challenges in Multimodality . . . 9
  - 2.2.4 Multimodal Data Processing . . . 9
- 2.3 Intelligent Multimodality in VR . . . 10
- 2.4 Summary . . . 11

3 Intelligent Modular Multimodal Framework Design **12**
- 3.1 Proposed Framework Design . . . 13
- 3.2 Pipeline Conceptual Models . . . 15
- 3.3 Closing Remarks . . . 16

4 Framework Prototype Implementation **17**
- 4.1 Pipeline Implementations . . . 17
  - 4.1.1 Hybrid Fusion Pipeline Implementation . . . 17
  - 4.1.2 Late Fusion Pipeline Implementation . . . 18
  - 4.1.3 Model Fine-Tuning . . . 19
  - 4.1.4 Module Implementation Summary . . . 20
- 4.2 Test Environment . . . 20
  - 4.2.1 Test Scene and Action Space . . . 20
  - 4.2.2 Automated Dataset Generation . . . 21
- 4.3 Hybrid Fusion Server . . . 23
- 4.4 Hardware and Software . . . 23
- 4.5 Closing Remarks . . . 23

5 Evaluation of the Pipelines **25**
5.1 Experimental Setup . . . 25
5.2 Results . . . 26
5.2.1 Ablation Study . . . 26
5.2.2 Missing Data . . . 27
5.2.3 Textual Impact . . . 28
5.3 Discussion . . . 29
5.4 Closing Remarks . . . 29

6 Conclusions and Future Work **30**
6.1 Conclusions . . . 30
6.2 Research Question Analysis . . . 31
6.3 Future Work . . . 31

References **32**

A Test Data **37**

List of Figures

3.1 Example of a possible interaction in the proposed VR context 13
3.2 Diagram of the proposed design for the prototype 14
3.3 Diagram of the hybrid fusion pipeline conceptual model 15
3.4 Diagram of the late fusion pipeline conceptual model 15

4.1 Example of a dataset entry being processed by the late fusion pipeline 18
4.2 Diagram of the image module's process 19
4.3 Diagram of the gaze module's process 19
4.4 Three screen captures of different possible configurations of objects, number shapes of the objects, and lighting of the scene 21
4.5 Example of an entry in the generated dataset 22

5.1 Column chart of both hybrid and late fusion pipelines' precisions with all modules 26
5.2 Column chart of the ablation study's precisions for the hybrid fusion pipeline . . 27
5.3 Column charts of the precisions when testing data is incomplete, modality-wise; hybrid fusion (H) and late fusion (L) 27
5.4 Column charts of the precisions when testing the textual data integrity; hybrid fusion (H) and late fusion (L) 28

List of Tables

2.1 Different modalities and their characteristics 5
2.2 Modality vs. Characteristic Mapping (1–5 Scale) 7

4.1 Comparison of implementation details for hybrid and late fusion pipelines 20

A.1 Precisions (0-1) used in basic fusion comparison 37
A.2 Precisions (0-1) used in hybrid fusion pipeline's ablation study 37
A.3 Precisions (0-1) used in missing data study 38
A.4 Precisions (0-1) used in text data comparison 38

Abbreviations

AI	Artificial Intelligence
CNN	Convolutional Neural Network
DL	Deep Learning
MLP	Multi-Layer Perceptron
HCI	Human Computer Interaction
ML	Machine Learning
NN	Neural Network
RNN	Recurrent Neural Networks
SVM	Support Vector Machine
VR	Virtual Reality
XR	Extended Reality

UN Sustainable Development Goals

The United Nations Sustainable Development Goals (SDGs) provide a global framework to achieve a better and more sustainable future for all. It includes 17 goals to address the world's most pressing challenges, including poverty, inequality, climate change, environmental degradation, peace, and justice.
This project, particularly the conclusions about the framework theorized and implemented, aims to contribute to goals: 4 - Quality Education, 9 - Industry, Innovation and Infrastructure and 10 - Reduced Inequalities. Applications made with the proposed pipeline could contribute to better and more advanced interactions both in industry and educational applications, as well as better accessibility alternatives.

Chapter 1

Introduction

The rapid advancement of Virtual Reality (VR) technologies has allowed new immersion opportunities to be explored and developed. As with many technologies, a focus on human interaction is primary, so research into user interaction and its nuances is ever important. Traditional input methods are often particularly good for one specific task but require the user to adapt to and learn them, creating limitations that cannot easily be solved. Multimodal approaches, which combine multiple input methods like gestures, speech, gaze, and touch, often allow for a more natural and immersive experience. These modalities can be fused by incorporating intelligent processes like Machine Learning (ML) models and Deep Learning (DL) algorithms, allowing the resulting combination to become a seamless and contextually-aware multimodal system. This dissertation explores the concept of intelligent multimodal 3D interactions in VR by designing and prototyping a framework that integrates a specific combination of modalities.

1.1 Context and Motivation

Multimodal systems draw on the Gestalt-like properties[1] of combining modalities to create more immersive and natural interfaces. This process can be cumbersome and complex, but recent advances in Artificial Intelligence (AI) technology have proven effective at advancing each modality individually and streamlining their fusion. Multimodal fusion remains a field with many open challenges, but when correctly implemented, it leads to drastically improved VR interactions, making them more immersive and engaging [41].

1.2 Problem

Despite the significant progress in VR technologies, a system that takes the advances of multimodal fusion and applies it in a flexible manner to VR environments has yet to be developed. Existing frameworks are either highly modular but not well suited for the immersive constraints of VR, or are built for VR but remain architecturally inflexible, supporting only a predefined set

[1] Psychology theory that proposes that the whole is greater than the sum of its parts

of modal inputs. This thesis addresses this knowledge gap by proposing a framework that offers both modularity and integration with VR environments.

1.3 Research Statement

This research, therefore, proposed the design, implementation, and evaluation of an intelligent, modular extensible multimodal intelligent framework for VR, envisioned as a low-investment alternative for developers. The core of this proposal is that a system architected for modularity, where input modalities can be dynamically added, removed, or replaced, will offer a versatile and accessible, "plug and play" tool for VR developers. This dissertation will investigate the feasibility of this approach by creating a functional prototype and evaluating its performance and flexibility against a traditional, non-modular alternative.

1.4 Research Questions

The research questions presented below aim to guide the project to explore and investigate the feasibility of an intelligent modular multimodal framework as a tool for developers:

- Research Question 1 (RQ1) - In what ways can AI be used to create a multimodal system for VR environments?
- Research Question 2 (RQ2) - What challenges appear in implementing modularity in an intelligent multimodal framework?
- Research Question 3 (RQ3) - What challenges are involved in applying an intelligent modular multimodal framework to the context of virtual reality?

1.5 Objectives

To tackle the identified problem and answer the research questions, this dissertation sets out to achieve a list of objectives. Firstly, design a novel framework for an intelligent and modular multimodal interaction system, specifically architected to be adaptable to the demands of VR environments. Then, implement a functional prototype of this hybrid fusion framework, using a combination of visual, gaze and speech inputs as a proof-of-concept. In order to serve as a benchmark for comparison, a non-modular and specific, late fusion pipeline is also to be developed. Both frameworks are then measured on precision, resilience to incomplete data, and performance impact of specific modalities, in the process quantifying the trade-offs inherent in the modular design. These findings are to be analysed in order to reach a conclusion on the feasibility of a flexible, AI-driven interaction tool to be used by VR developers.

As for use cases, this framework is planned to serve as the basis for a low-investment alternative for developers who wish to implement multimodality in their VR projects. It could serve

as an aid in the prototyping phase of development, where, for example, a developer could quickly implement and test an interaction where a user looks at a tool (gaze), states an action (speech), and performs a gesture, all without needing to build the underlying input processing logic from the ground up. In architectural design or 3D modeling applications, a designer could rapidly prototype workflows where virtual objects are manipulated with gestures while voice commands change attributes, allowing for quick iteration on user experience concepts.

The primary advantage of the proposed methodology is its accessibility and flexibility. Unlike conventional systems that require significant upfront investment, this framework allows for a more "circuit breadboard" like approach, with a lower entry barrier, where a developer can more easily experiment with sophisticated multimodal interactions. Its modularity also works towards rapid testing of different modality combinations to find what works best for a specific application. This design philosophy comes with inherent trade-offs, wherein the focus on versatility and ease of use may come at the cost of performance. When compared to a more bespoke solution to a given problem, this modular alternative could imply greater computational overhead and latency.

1.6 Document Structure

In addition to this introductory chapter, this dissertation presents in chapter 2 a study on the current state of the methodologies and technologies mentioned, particularly the intersection of AI, multimodality, and VR. Chapter 3 presents the concept design for the research at a high-level, overviewing the two frameworks to be developed, and in chapter 4, these designs are implemented. This includes the development of the Unity test environment, the process for creating a custom training dataset, the specific details in the implementations of the ML models used in both pipelines. The results of the evaluation are presented in chapter 5, where the experimented results are compared and analysed. Both hybrid and late fusion models are compared across several metrics, including an ablation study to measure the impact of each modality on the hybrid fusion pipeline. These results are then discussed and used to answer the research questions presented in this chapter. Chapter 6 concludes the dissertation by summarizing the key findings, highlighting the challenges, sucesses, limitations while proposing possible avenues for future research on this topic.

Chapter 2

State of the Art

The previous chapter established a clear problem: the lack of a system that is both modularly flexible and natively designed for VR. To begin solving this, we must first understand the tools and techniques available. This chapter, therefore, surveys the current state of the art to build a comprehensive understanding of intelligent multimodal interaction. Our exploration is divided into three parts. Section 2.1 includes an overview of the fundamental input modalities, from speech and audio to gesture and haptic. With these components identified, Section 2.2 investigates how they can be fused, reviewing different AI-driven models for combining their data. Finally, Section 2.3 connects this knowledge back to our primary goal by analyzing how these intelligent fusion strategies are currently being used—and where they fall short—in modern VR applications.

2.1 Aggregation and Analysis of Various Modalities

Since "modality" is a loose term, this dissertation requires a more thorough taxonomy to describe different types of user-system interactions. Table 2.1 was elaborated to map various modalities to specific and relevant characteristics. To illustrate the need for this organization, consider the act of writing on a standard keyboard. This input device consists of a broad multimodal interaction that includes the haptic sense of pressing keys, the auditory of the keys' sounds and the visual feedback of the characters pressed appearing on a screen. However, the keyboard is not commonly referred to when talking about multimodal interfaces as there is a hierarchy in its constituent modalities. Only the haptic aspect of the pressing the physical keys is considered the primary modality, while the others are auxiliary/secondary, effectively categorizing the keyboard as a unimodal interface. Therefore, using this term is assumed conditional, based on an interaction method's modal constitution. Among the chosen characteristics for categorization:

- **Data Direction** is separated into Human-Computer and Computer-Human, and it is used to specify if the modality captures data from the user to be used by the system or if it takes data from the system and presents it to the user;

- **Level of Data Processing** is used to describe a modality's separation from raw sensor data, as well as its involvement with different technologies;
- **Intent** details whether a given modality can transmit explicit user intent.

Table 2.1: Different modalities and their characteristics

Type / Modality	Data Direction	Level of Processing	Intent	Representative Papers
Speech	Both	High	X	[19, 24, 39, 51]
Haptic	Both	High	X	[6, 15, 50]
Text	Both	High	X	[6, 9, 52]
Hand gesture	H $\rightarrow$ C	High	X	[13, 24, 31, 38, 51]
Body gesture	H $\rightarrow$ C	High	X	[1, 4, 26, 31, 44]
Facial gestures	H $\rightarrow$ C	High	X	[31, 33]
Gaze	H $\rightarrow$ C	High	X	[44, 51]
Brain - Computer	H $\rightarrow$ C	High	X	[40]
Skin Stretch and Deformation	H $\rightarrow$ C	High	X	[5]
Object manipulation	Both	Low	X	[9, 19]
Tongue	Both	Low	X	[16, 23, 29]
Inertial sensor	H $\rightarrow$ C	Low	X	[26]
Muscle Activity	H $\rightarrow$ C	Low	X	[13]
Exhalation	H $\rightarrow$ C	Low	X	[47]
Image / Vision	Both	High		[1, 6, 9, 20, 23, 24, 26, 32, 33, 39, 52]
Emotion	H $\rightarrow$ C	High		[33]
Galvanic Skin Response	H $\rightarrow$ C	Low		[20]
Heart rate	H $\rightarrow$ C	Low		[40]
Audio	C $\rightarrow$ H	High		[1, 6, 32, 39]
Taste	C $\rightarrow$ H	Low		[23, 29]
Smell	C $\rightarrow$ H	Low		[1, 23]

Table 2.1 includes modalities that can be considered lower-level (atomic) or higher-level (composite). The choice to include both is because, by omitting them, specific details would be lost, as lower-level modalities can be part of higher-level ones but are also specific enough to stand alone and have prolific uses by themselves. To illustrate this, consider the interplay of audio and speech. Audio, as raw sound input, is a lower-level, atomic modality, and can be used by itself. However, audio can be combined with text processing to create a higher-level modality of speech recognition. Although speech recognition relies on audio processing, keeping them as separate entries preserves the unique utility of each modality. Additionally, some modalities, such as facial gestures, exhalation, and others, can be thought of as being bidirectional in their data direction categorization due to the possibility of using other means to return this information to the user (e.g., using a screen display to show the user an image of a face making an expression). This, however, is considered to rely on modalities other than the one, so these cases are categorized only as Human-Computer (H $\rightarrow$ C).

2.1.1 Filtering of Found Modalities

The modalities presented in Table 2.1 are ordered in such a way that prioritizes their aptness for an intelligent, multimodal, human-to-computer system. The capacity to convey intent is the most essential characteristic, given that if the user cannot control their input, the modality serves no purpose to the system. Following that is the level of data processing it holds, which is related to the modality's complexity and, by extension, the nuances an intelligent method like DL and ML can retain from analyzing its data. Including data direction ensures that the data from the modality does flow from the human to the computer, but also its flexibility if the data flows in both directions.

In short, relevant modalities near the top of the table carry information from the user to the computer, are complex enough to garner interest in being explored with an intelligent method, and can convey user intention. The modalities that fit the criteria are speech, haptic (tactile sensation), text, hand gesture, body gesture, facial gesture, gaze, brain-computer, and skin stretch and deformation.

2.1.2 Introduction to Common Modalities

Speech recognition has been explored for quite some time, existing in the workings of commonly used technology, like digital assistants (Apple's Siri [46], Microsoft's Cortana [30], and others) [18]. Speech is one of the main ways we communicate information [12], natural language being especially good at conveying abstract concepts and relations [41].

Text recognition is a very explored field, the basis for wide-spanning bodies of knowledge like natural language processing. It subsections into many processes, like text classification, word embeddings, sequence labeling, and eventually transformers and Large Language Models (LLM), as well as the various methodologies to deal with each of them [22]. Currently, the vanguard of this modality is LLMs like OpenAI's ChatGPT [35].

Haptic systems rely on the complex tactile nature of the skin and muscles to create the illusion of objects/surfaces that are not there. Existing systems like [7, 10, 27] display a variety of approaches to implementing this modality. These technologies have to consider a multitude of variables like degrees of freedom, input position resolution, maximum force/torque, and many others [15].

Gesture recognition can be divided into body, face, and hand recognition, each being its own microcosm of potential research. Body recognition tends to be less precise than the other two and is often used to analyze posture, gait, and overall body shape. Facial recognition is mainly used for subtle emotion recognition and identification. Hand gesture recognition is possibly the most used of the three for exploring interaction and input methodologies, given the high expression quality of the hands, finding most use in Extended Reality (XR) systems [34, 41].

Gaze recognition can be beneficial, as vision is the sense most responsible for information gathering [28]. Tracking where the eye is looking, or more interestingly, where the user's visual

focus is, where it lingers, and where it returns more often, can allow a system to, in many cases, in addition to other modalities, more accurately decipher user intent.

2.1.3 Characterization of Candidate Modalities

Using what was found during the research, and by focusing on promising modalities to explore, 5 pragmatic criteria were used to evaluate the relevance of using a given modality in this research. It is important to note that the subsequent analysis represents a qualitative assessment based on the literature reviewed for this dissertation. It serves as a heuristic tool to guide the selection process, not as a definitive, peer-validated metric.

The modalities were rated against each criterion on a 1-to-5 scale, where a score of 1 indicates significant barriers, and a score of 5 represents high suitability and accessibility for a VR research prototype.

These characteristics are:

- **Ease of access and implementation**: How easy it is to acquire and implement the equipment and resources required to gather and process data;
- **Maturity of the field**: How much research and time has been spent on exploring the field;
- **Ease of data gathering**: How much existing available data there is, and how easy it is to acquire more;
- **Accessibility**: How user-friendly and accessible the modality is to any given user;
- **VR propensity**: How adaptable and compatible a modality is.

Table 2.2: Modality vs. Characteristic Mapping (1–5 Scale)

Modality	Ease of Access	Maturity	Data Gathering	Accessibility	VR Propensity	Total Score
Speech	5	5	5	4	5	**24**
Image	4	5	5	5	5	**24**
Gaze	3	5	5	5	5	**23**
Body Gesture	4	4	5	4	5	**22**
Haptic	5	5	5	4	3	**22**
Text	5	5	5	5	1	**21**
Facial Gestures	3	4	4	5	4	**20**
Hand Gesture	3	4	4	4	4	**19**
Brain–Computer	1	3	4	5	5	**18**
Skin Stretch & Deformation	1	1	2	5	3	**12**

Table 2.2 presents this evaluation. Modalities on the table are ordered by the sum of the values, speech showing the greatest score, and skin stretch and deformation the least score. The results highlight that modalities like speech, image, and gaze are particularly well-suited for this dissertation's objectives.

2.2 Multimodal Fusion

Analyzing each modality individually can give insight into its nuances, but it does not cover its synergistic abilities. For this purpose, further characterization and analysis is required. These modalities follow a "Gestaltian" principle [41], where multimodal systems are often greater than the sum of their constituents. Any modality has its requirements, pros, and cons; combining modalities can make up for each other's weak points and create a new experience altogether.

2.2.1 Combining Modalities into Multimodalities

In the field of multimodality, there are many approaches when combining said modalities [41]:

- **Exclusive** multimodality refers to when modalities are used at different times for different purposes. Example: selecting objects with gestural interaction and entering unrelated information with dictation;
- **Concurrent** multimodality happens when modalities are independent and used in parallel. Example: browsing pictures with hand gestures while performing a voice search;
- **Alternate** multimodality occurs when modalities are sequential but linked. Example: pointing an object with a controller and giving a voice command to copy it;
- **Synergistic** multimodality occurs when modalities are used in parallel and linked. Example: pointing and giving a related speech command at the same time.

Using this taxonomy, the synergistic type of multimodality seems to be the best to explore, given that it allows for an AI model to parse all modalities at once and derive a single meaning from them, given that all modalities involved are used towards a single intention.

2.2.2 Introduction to common multimodal combinations

Image and text are commonly combined exclusively together in a bimodal manner. A frontier of this development seems to be Visual Language Models (VLM) [9] like GPT-4V [37], which combine LLMs with image processing. This combination tends to achieve both an enhanced problem-solving capacity of these models, as well as AI-generated images provided by generative AI like Google's Gemini [49], or OpenAI's DALL-E [36].

Gesture and speech are often used together, complementing each other very well. Speech's limited ability to explain distances, directions, and spatial relations is solved by gesture recognition, which generally cannot parse complex user intent with the simple commands it works with. Gaze recognition can be used as a substitute for each of these, depending on the case study, or in more ambitious situations, in combination with them, to achieve an even higher degree of intent and pattern recognition [41].

2.2.3 Challenges in Multimodality

Multimodality as a discipline is very promising, but it is not without its challenges. Being a combination of various technologies and data, it suffers from the *curse of dimensionality* [28]. Adding to the problem, modalities vary in how they are collected, at what times, and in the structure that the sensor data is formatted.

Regarding how their data is gathered, there are various approaches to sensor making and location in the use scenario in most modalities. An example of this would be the positional / orientational sensors of a VR headset: depending on the system, the approach can either be outside-in, meaning that the sensors are external, fixed points in the space, that receive the headset's data and send it to the system; or inside-out, meaning that the cameras are embedded in the headset. This seemingly minor change in the scope of the modality requires wholly different technologies and frameworks to work.

Sensor data can also be collected at different moments in the scenario. Depending on the modalities at hand, the multimodal system can be asynchronous or synchronous, meaning that the various data acquired are either processed at once or at different times, most likely due to hardware requirements/restrictions. For example, in a VR headset, the modalities at hand (any combination of image, sound, haptic, proprioceptive, and others) are required to obey a "window of integration"; if they are too out of sync, for any hardware or software reason, they can heavily impact the user's experience, potentially leading to physiological reactions like discomfort and nausea [28].

When it comes to data standardization, there are currently no easy solutions. Audio data will always be categorically different from textual data, so their formatting often falls under the program's responsibility. Approaches to multimodal fusion and data handling are explored further in the document.

In short, as data increases in volume, dimensionality, and variety, the processing complexity grows exponentially. This exponential nature means that modalities have to be carefully chosen for any given project, as the challenges introduced by each additional input can quickly outweigh its benefits.

2.2.4 Multimodal Data Processing

Regarding processing data in multimodal systems, also known as multimodal fusion, various approaches can be taken [43, 48]. Model-based approaches are tailored to a single system. The fusion is embedded into the model, optimizing it for its specific use case. The drawback of this approach is that no other use case can use this system.

Model-agnostic methods do not consider any specific ML model or architecture. They are focused on treating the data, through feature treatment like concatenation and statistical methods like averaging, into a format that can be later processed by any one model [32]. Using an analogy, a "Swiss knife" is a tool that, although versatile, is not as precise and efficient at any one specific task as the tools it attempts to recreate. Model-based approaches are standard, specific tools like

a "knife" or a "corkscrew," while model-agnostic approaches are "Swiss knives." To further delve into model-agnostic methods, these can be divided into three main categories of early, late and hybrid fusions:

In early fusion, raw data, or low-level features from multiple sources are combined into a unified representation before being fed into a model. This method allows the model to learn the relationships present in the data directly. It does, however, require that all modalities be synchronized, which is not always the case, and it often creates high-dimensional inputs, which are requirements that only specific systems and use cases can support. An example of early fusion would be concatenating pixel-level features of an image and word embeddings from text into a single feature vector [33, 39].

Late fusion is the opposite; raw data from each modality is often processed individually by different means, outputting their results (probabilities, predictions, and others), which are then combined in a decision-making stage. This method allows for asynchronous multimodality, fine-tuning for each modality, and making decisions, even if an incomplete set of inputs is given to the model. This individuality, naturally, comes at the cost of data relation comprehension, as the model used can not make its own decision from the original, untreated data. An example of this method could be analyzing audio and video streams separately using distinct models and then combining their results [2].

Hybrid fusion tries to conciliate these two concepts. Methods that follow this structure can refine raw data into high-level features, which are then merged into a unified representation for further processing. This method has the benefit of allowing each modality's strengths to be reinforced and capturing complementary and synergistic information that would not be apparent had the modalities been completely processed separately. An example is extracting pitch and rhythm from an audio source, semantic meaning from a text, and spatial features from a video and then inputting these newly treated features into a model for classification [6, 11].

2.3 Intelligent Multimodality in VR

The intersection of multimodality, AI, and extended reality is a field that has only recently been having significant activity, with most papers published mainly in the past four years (2020-2024), peaking in 2023 [8]. This trend shows that, despite the constituent fields and their pair combinations being researched, the combination of the three is still very unexplored and brimming with potential.

The main idea of this combination is that XR experiences can be significantly enhanced with better applications of multimodality and that introducing ML and DL methods can bridge them together by making the implementation of complex and intricate systems easier and more straightforward [8].

Most studies on this intersection tend to focus on two modalities, making for a more bimodal study, image and text being the most common combination. Few studies that fit this category venture into a "true" multimodal approach [8], showing a sensible knowledge gap in the field.

2.4 Summary

This chapter briefly reviews current research and developments on intelligent multimodal systems, focusing on their applications in VR scenarios. It creates a simple taxonomy for distinguishing which known modalities would be more relevant to explore. It also explores multimodal fusion strategies and their nuances, specifically when fusion should happen in the model, and showcases common challenges in this process. This uncovered knowledge reveals the complexity and potential of intelligent multimodal VR systems.

Chapter 3

Intelligent Modular Multimodal Framework Design

The preceding chapters have established a gap in the field: Chapter 1 identified the lack of a system that is both modularly flexible and optimized for VR, and Chapter 2's review of the state of the art revealed the potential of using a hybrid fusion model to address this challenge. With this analysis, this chapter is dedicated to designing a solution, an intelligent modular multimodal framework.

The objective is to have a tool that allows a developer of any given application in a VR environment to choose an arbitrary set of modalities and use the model to interpret them, outputting an action from a list of possible actions involving a set of interactable items. This chapter lays out the design and conceptual model for this approach, detailing the design of a flexible hybrid fusion pipeline. As a point of comparison, a non-modular late fusion pipeline is also architected. By comparing these two designs, it is possible to explore the requirements and trade-offs of building modularity into an intelligent interaction system.

To provide a more concrete idea of this concept, consider two developers building different VR applications but wanting to use the same underlying input system.

- **Developer A** is creating an architectural design tool. They want users to combine speech and hand gestures to manipulate objects (e.g., pointing at a wall and saying, "Make this blue");
- **Developer B** is developing a medical training simulation. They need a silent, precise interface, so they combine gaze for object selection and controller input (touch) for command execution.

A truly modular framework should support both developers. The core pipeline, especially the Multi-Layer Perceptron (MLP), a Neural Network (NN) whose nodes are fully connected, must be abstract enough so that it can serve as a foundation for both systems, despite the use of entirely different sets of modalities. The developers would only need to integrate the specific modules relevant to their project.

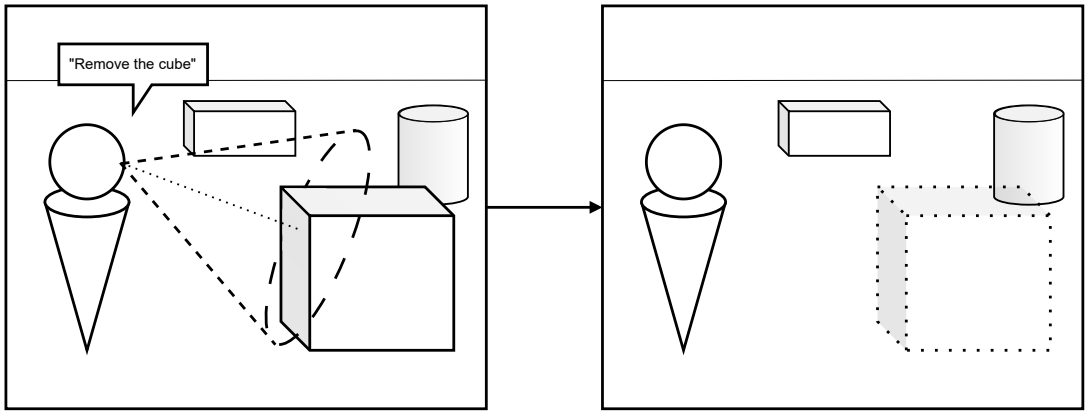


Figure 3.1: Example of a possible interaction in the proposed VR context

For this project and as a proof of concept, this model is instantiated to the context of VR (in section 2.1), where the objective is to create a tool that takes visual, speech, and gaze data from the user in a virtual reality environment, and outputs an action that the user applies to an object in the scene, such as presented in Figure 3.1. The context of VR also imposes certain performance metrics on the project, as VR environments require specific parameters in order to provide immersive and non-jarring experiences for the user. These parameters include response time, system throughput and resource utilization, among others [42].

Recalling the conclusions of Chapter 2, the choice of modalities (visual, speech, and gaze) is informed by the criteria present in Table 2.2. Multimodal datasets are scarce, especially depending on the specific modalities and when more than two modalities are involved.

3.1 Proposed Framework Design

To achieve modularity, one of the main goals of the project, a hybrid approach is proposed. As argued in the previous chapter, this method, in contrast to early and late fusion approaches, allows the modalities to be interchangeable, processing data just enough to fit into a compatible format while maintaining the rich correlated information for the MLP to process and interpret correctly. A high-level design of this overall framework is shown in Figure 3.2

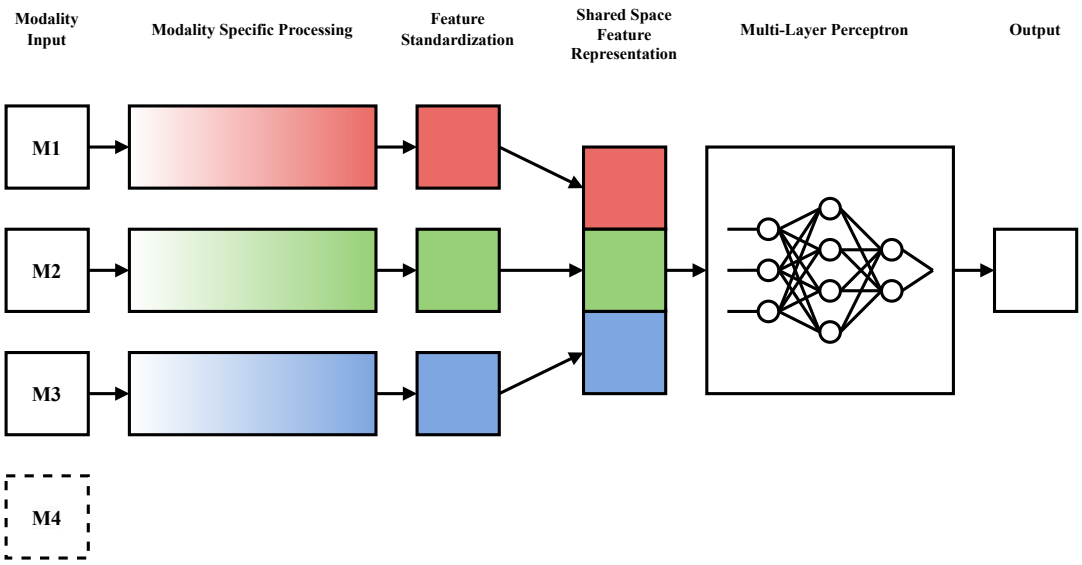


Figure 3.2: Diagram of the proposed design for the prototype

In the more detailed design of the hybrid fusion pipeline, shown in Figure 3.3, each modality's data is transformed into an intermediate representation and then concatenated into a single data "package". This "package" is then fed into a neural network trained to map this combined data to the user's final action. Doing this makes it so that, while a specific combination of modalities requires the network to be trained on that corresponding input format, the underlying pipeline and the MLP remain structurally unchanged. This allows a developer to add, remove, or swap the pre-processing modules and then simply retrain the network, without having to re-engineer the entire fusion logic from scratch for each new use case.

As mentioned previously, to provide a point of comparison to this hybrid fusion framework, a non-modular, late fusion alternative was also conceptualized. The key difference in this method, illustrated in Figure 3.4, lies in the complete treatment of modal data where the output of each module is one or more of the final components of the complete output. The simpler, tailored network in the pipeline receives these already decided inputs and deduces the final output based on which modality they came from. More in-depth details about both pipelines implementations are provided in Chapter 4.

This comparison is essential to retrieve insights into the strengths and possible shortcomings of each approach. The hybrid fusion approach prioritizes modularity, allowing the model to learn complex inter-modal relationships, potentially at the cost of longer training times. Conversely, the late fusion approach prioritizes specificity and nuance, employing specialized modules to process each modality individually, resulting in a more rigid, tailored structure that completely sacrifices modularity.

3.2 Pipeline Conceptual Models

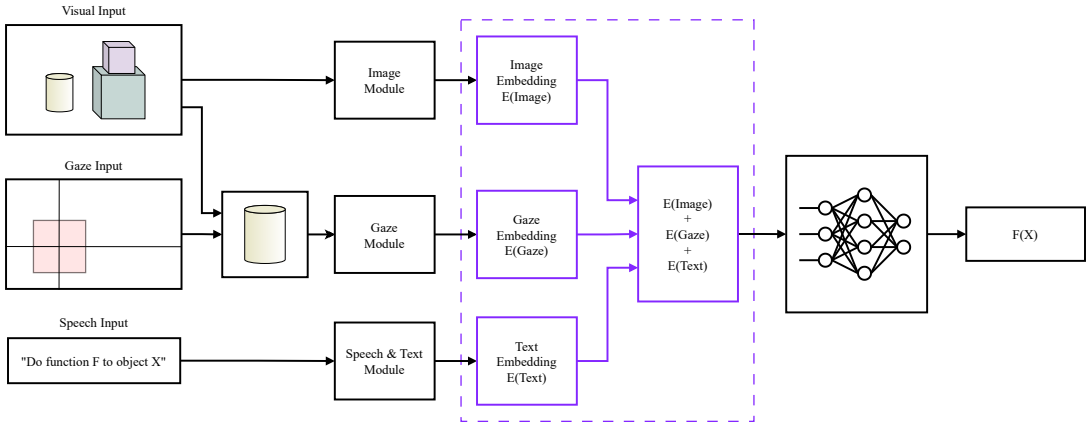


Figure 3.3: Diagram of the hybrid fusion pipeline conceptual model

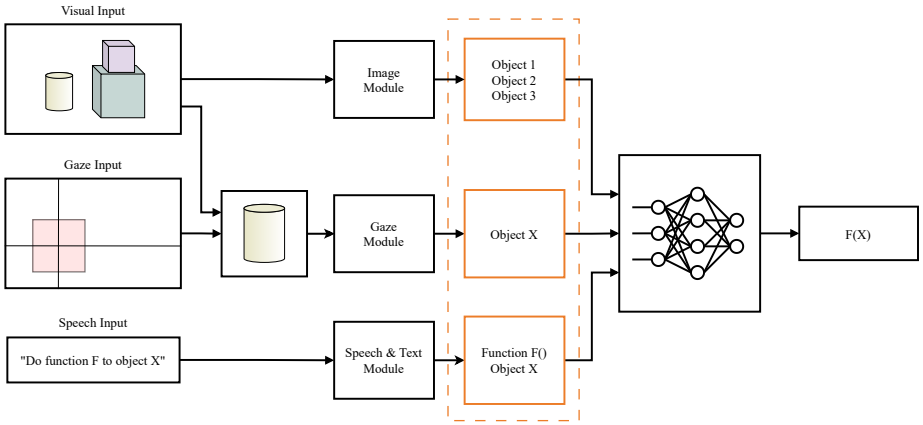


Figure 3.4: Diagram of the late fusion pipeline conceptual model

Figure 3.3 also shows a case study using image, gaze, and speech modalities as an example of the hybrid fusion pipeline's workings. Each module, upon receiving its corresponding raw data, applies any one method to standardize it into a coherent format that can combined with the other modules' outputs. Specifically, each modality's input is processed by its specific pre-processing module, where *E(image)*, *E(gaze)*, and *E(speech)* are the internal representations (known as **embeddings**) of each respective modality. These are then combined and inputted into the final MLP model, which outputs a *Function(object)* action, where the specific result is one element of the list of possible user actions. This pipeline's training process focuses on teaching this MLP to understand the relationships between the embedded inputs and the final user intent.

In contrast, the late fusion pipeline's implementation, as shown in Figure 3.4, relies specifically on the case study of this project, using the modalities image, gaze, and speech, processing them using modality-specific modules. These modules, compared to the hybrid fusion pipeline's, do a complete treatment of the raw data, converting it into valid parts of the final action output.

This difference is illustrated using the different dashed-line boxes in both Figures 3.3 and 3.4, highlighting the modules' outputs, before being fed into the MLP. The visual data (image) is passed through a locally fine-tuned object detection NN model, which outputs the classes of the virtual objects in the test scene. The speech data, simplified to textual data, is interpreted by a fine-tuned sentence transformer NN model, which outputs both possible object classes and functions. The gaze data is a set of screen coordinates applied to the visual data, cropping a focused area of the image, which is passed through the same fine-tuned model as the visual module, outputting the object classes present in that focused area (further elaborated in section 4.1).

These human-readable high-level outputs (e.g., image-output=[cube, sphere], gaze-output=[cube], speech-output=[delete, cube]) are then fed into the final MLP model, which is locally trained on the possible function and object combinations, resulting in a final *Function(object)* action.

3.3 Closing Remarks

This chapter presents a conceptual blueprint for an intelligent multimodal framework aimed at addressing the flexibility gap in VR development. It proposes a hybrid fusion architecture to serve as the core of this framework, designed for modularity by standardizing diverse inputs into a shared feature space for a central MLP to process. To create a basis for rigorous evaluation, it also architects a late fusion pipeline as a non-modular and specialized alternative that prioritizes task-specific performance over flexibility. These two opposing designs establish a clear experimental setup for investigating the fundamental trade-offs between modality adaptability and model accuracy.

With these conceptual models now established, the next logical step is to move from design to execution. The following Chapter 4 details the practical realization of these pipelines. It describes the development of the Unity test environment used for data gathering, explains the process of creating a custom dataset to train the models, and provides the specific implementation details of the neural networks that power both the hybrid and late fusion systems.

Chapter 4

Framework Prototype Implementation

Using the conceptual blueprints for both the modular hybrid fusion, and the specialized late fusion pipelines established in Chapter 3, this chapter details their practical implementations.

First, it details the concrete implementations of the architectures, showing how the abstract designs of both pipelines were instantiated using specific ML models. Second, it presents the test environment and custom dataset used to train, validate and test these implementations. Finally, it outlines the deployment of the hybrid model developed into a local server, in order to assess its viability for live interaction.

4.1 Pipeline Implementations

This section translates the architectural designs of Chapter 3 into functioning code, detailing the specific models and techniques used to build both hybrid and late fusion pipelines.

4.1.1 Hybrid Fusion Pipeline Implementation

As mentioned in Chapter 3, the core principle of this design is to achieve modularity by having each input module convert its raw modal data into a standardized, information-rich format. Doing this, each module produces a 1D vector, which are then concatenated to form a single input for the final MLP, which is trained to interpret this combined representation, as seen in Figure 3.3.

Enumerating through the modalities, the image data is passed to Torch Vision's ResNet50 image classification model [17], which outputs a dense embedding vector, flattened into a 1D vector. The textual data is passed to a fine-tuned version of Sentence Transformer's "all-MiniLM-L6-v2" sentence embedding model [45], which, unlike in the late fusion implementation, only outputs the 1D vector, not a full classification. The gaze data is used to crop the image data into a smaller image corresponding to the area of interest, which then passes through the same treatment as presented in the image-data processing procedure, outputting a 1D vector. These specifications are further explained in section 4.1.4.

This approach implements the concept of a shared feature space shown in Figure 3.2. The modularity is preserved because any module that can output a 1D vector can be swapped in or out;

only the final MLP needs to be retrained on the new input shape, without needing to be remade in order to fit the new parameters.

4.1.2 Late Fusion Pipeline Implementation

In direct contrast to the hybrid approach, the late fusion pipeline was designed (as seen in Figure 3.4) to have each module perform specialized, high-level processing, outputting human-readable classifications that are components of the final action.

As for the modules, image data is passed to a fine-tuned version of YOLO's "yolov8s" object detection model [21], which outputs the shapes of the identified objects. Textual data is passed to a fine-tuned version of Sentence Transformer's "all-MiniLM-L6-v2" which outputs both the identified function and object. Gaze data is used to crop the image into a smaller one corresponding to the area of interest, which then passes through the same treatment as presented in the object detection procedure, outputting the shape of the identified objects in this focused area.

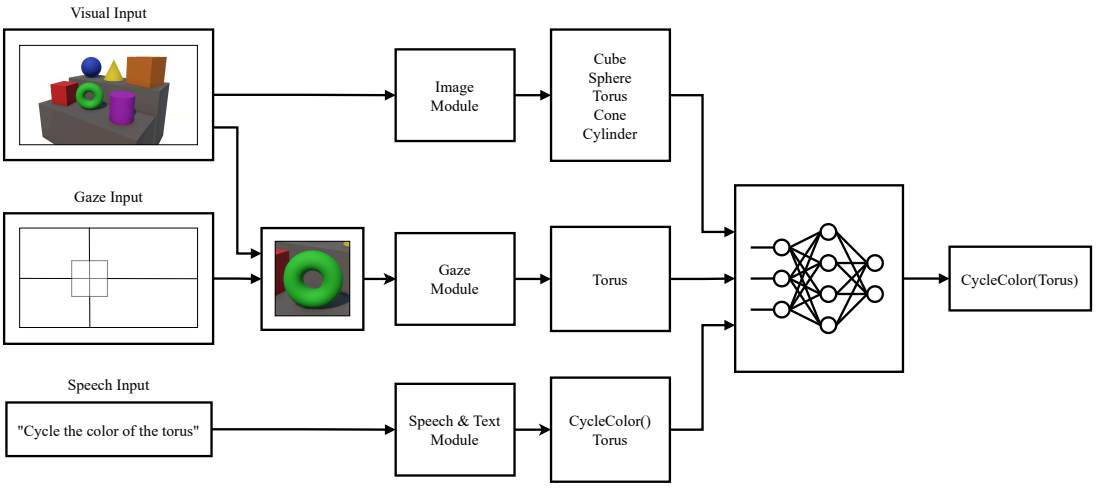


Figure 4.1: Example of a dataset entry being processed by the late fusion pipeline

An example of the late fusion pipeline's behavior is shown in Figure 4.1, where a sample entry of the dataset is used. In the figure, the image is processed by the image module, outputting all the present objects' classes; the speech, in the form of text, is processed by the corresponding module, and both function and target object class are extracted; the gaze coordinates are used to cut a small image of the target object, in this case, a torus, and this new image is processed by the gaze module, returning the focused object's class. After the trained MLP processes these values, the final action is 'CycleColor(Torus)'. Figures 4.2 and 4.3 aim to better illustrate the inner workings of the image and gaze modules, respectively.

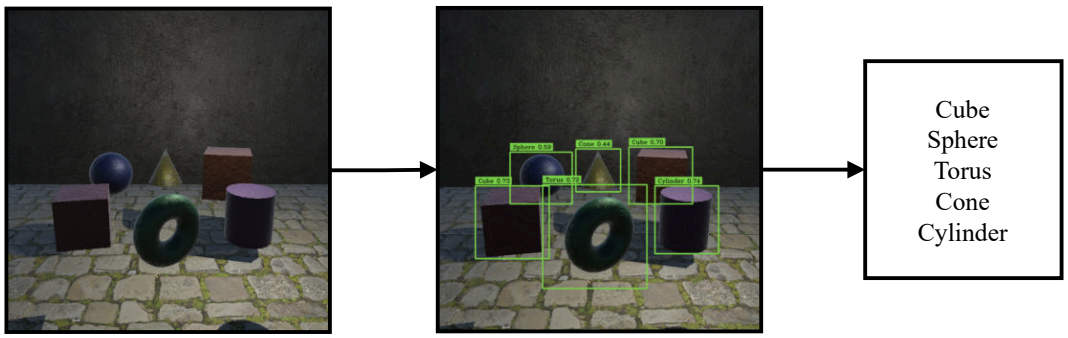


Figure 4.2: Diagram of the image module's process

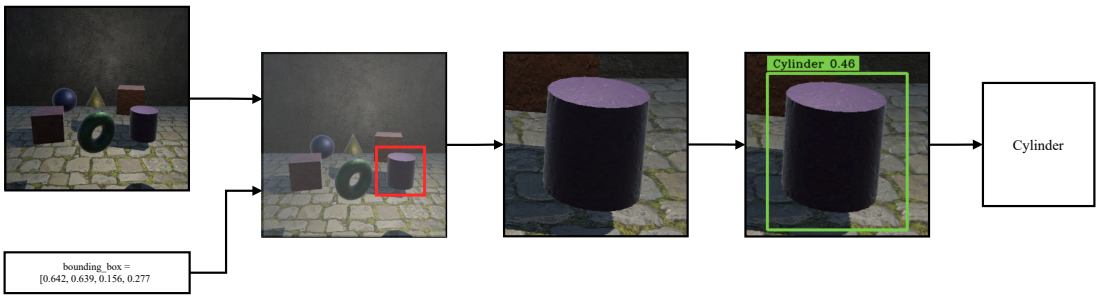


Figure 4.3: Diagram of the gaze module's process

4.1.3 Model Fine-Tuning

Additionally, besides the trainings of the final MLPs, some of the models used had to be fine-tuned. YOLO's "yolov8s" model is pretrained only for a particular list of classes (COCO 2017 dataset, with 80 classes [25]), and does not correctly recognize the test environment shapes (torus, cone, etc.). Its fine-tuning involved freezing the original model's first 10 layers, and subsequently training the model on a training dataset, as explained in the following section 4.2.2.

The language processing model, Sentence Transformer's "all-MiniLM-L6-v2", also needed to be fine-tuned, or else it struggles to correctly separate the semantic meanings of more nuanced, lexically similar functions like *MakeBigger* and *MakeSmaller*, easily confusing them for each other. Its fine-tuning used a specialized dataset to teach semantic similarity and dissimilarity. For any given command, each entry also included examples of phrases that meant the same thing ("positives") and phrases that meant something different ("negatives"). This training process teaches the model to group semantically similar commands closely together in its internal vector space, while pushing dissimilar commands apart. This ensures that all variations of a single command are understood as the same concept, creating a robust and unambiguous mapping of our command set.

Fine-tuning was therefore essential to adapt these general-purpose models to the specific vocabulary and object shapes of the test environment.

4.1.4 Module Implementation Summary

To provide a clear comparison between the two distinct implementations, Table 4.1 summarizes the role and output of each module in both pipelines.

Table 4.1: Comparison of implementation details for hybrid and late fusion pipelines

Component	Hybrid Fusion Pipeline (Modular)	Late Fusion Pipeline (Specialized)
Image Module	Processes image into a dense feature embedding using a pre-trained ResNet50 [17], and flattens it into a 1D vector.	Detects all objects and outputs their class labels using a fine-tuned YOLOv8 [21].
Text Module	Converts text into a semantic embedding using a fine-tuned all-MiniLM-L6-v2 [45], in the format of a 1D vector.	Classifies text into Function and Object labels using a fine-tuned all-MiniLM-L6-v2 [45].
Gaze Module	Crops image and generates a focused feature embedding using ResNet50 [17], and flattens it into a 1D vector.	Crops image and outputs the focused object's class label using YOLOv8s [21].
Fusion Stage	Concatenates all embeddings into a single large vector. A complex MLP learns relationships from this low-level data.	Concatenates class labels. A simple MLP makes a final decision based on this high-level, pre-classified data.

As Table 4.1 shows, the difference between both pipelines' modules lies in the level of abstraction of their outputs. The hybrid pipeline works with low-level, dense embeddings, reduced into 1D vectors, delegating the complex interpretation task to a single, powerful MLP. Conversely, the late fusion pipeline's modules each fully process their inputs, leaving only a simple decision-making task for its final MLP. This architectural choice is one of the roots of the performance and flexibility trade-offs that are evaluated in Chapter 5.

4.2 Test Environment

Having detailed the implementation of the pipelines, this section describes the controlled environment and custom data created to train and test them. This dataset was designed specifically to provide the necessary inputs (i.e., visual, text and gaze) and to benchmark the performance of both pipelines on well-defined tasks.

4.2.1 Test Scene and Action Space

A simple test scene was developed in the Unity engine[1], examples of it shown in Figure 4.4. The environment was designed to generate procedurally, consisting of:

- **Background:** A simple but textured background consisting of basic planes;
- **Objects:** An arbitrary number of objects, all of which can have one to 5 different shapes (cube, sphere, cone, cylinder, torus) and one of 20 different materials;

[1] https://unity.com/

- **Lighting:** Any number of 10 possible differently colored lights active in the scene.



Figure 4.4: Three screen captures of different possible configurations of objects, number shapes of the objects, and lighting of the scene

The full pipelines' outputs consist of an action, the composite of a function and an object, expressed as *Function(Object)*. The five possible functions used are:

- **Destroy(x):** Deletes object x;
- **MakeSmaller(x):** Reduces the scale of object x;
- **MakeBigger(x):** Increases the scale of object x;
- **ChangeMaterial(x):** Assigns a new, randomly chosen material to object x;
- **CycleColor(x):** Begins a coroutine that periodically cycles the color of object x, over a random period of time.

These functions were chosen due to various factors, such as their complexity, as being unary functions allows for easy parity between both pipelines. In contrast, more complex functions would have proven inefficient in implementation and usefulness. The late fusion pipeline must be crafted carefully to construct the final actions, especially compared to the relatively input-agnostic hybrid fusion pipeline. The inclusion of similar lexical functions (e.g., 'MakeSmaller()' and 'MakeBigger()') is a useful hurdle, as the models must adapt to it in their training to understand small/self-similar lexicons in possible future projects.

4.2.2 Automated Dataset Generation

As public datasets matching the specific multimodal needs for this exploration do not exist, a custom dataset was created. A C# script was developed in Unity that, for each entry, randomizes the attributes of the scene, adjusting the shapes and materials of scene objects, as well as the active lights, in order to ensure statistical diversity and prevent model overfitting.

This script positions the camera at a random distance from the target object, within given parameters, ensuring it faces the object. It takes a 1920 x 1080 pixel viewport screenshot and

saves it. In this process, it calculates the bounding boxes of the objects in the image and saves them to files named respectively. Then the script randomly selects one of many given natural language commands and chooses one of the five functions and one of the present objects to fill the command (e.g., "destroy the cube", "change the sphere's material").

In greater detail, the developed script selects the object's shape by uniformly sampling from a list of available shapes, indexed over the discrete interval $[0, N-1]$, where N is the total number of shapes. The material is chosen in a similar manner, using uniform sampling from the list of candidate materials, also indexed over $[0, M-1]$, with M being the number of materials. As for lighting, a random integer k is drawn from the interval $[0, \lfloor L/2 \rfloor]$, where L is the total number of lights, and k unique lights are randomly selected and activated. The camera position is determined by sampling a direction uniformly on the unit sphere and scaling it by a radius $d \sim U(\text{minDistance}, \text{maxDistance})$, where $U(a,b)$ is a continuous uniform distribution over the interval (a,b). The camera is then aimed at the object, with an additional random roll angle $\theta \sim U(0^\circ, 360^\circ)$ applied about its forward axis.

An extra consideration taken in the choosing of functions and commands that implement them was the size of the command. The objective is to simulate what a user of the VR application might naturally say, producing smaller, objective commands, possibly skipping important context, and using synonyms of the function instead of its 'correct' name.

The result is a dataset where each entry, as illustrated in Figure 4.5, includes an image path, a synthetic, natural language command, screen coordinates of the target object's bounding box's (the values are in a normalized 'xywh' format, where x and y are the coordinates of the center point of the box, while w and h are the height and width, respectively), and the ground-truth 'Function(object)' label. The datasets created were training, with 2520 unique entries, validation, with 510 unique entries, and testing, with 510 unique entries.

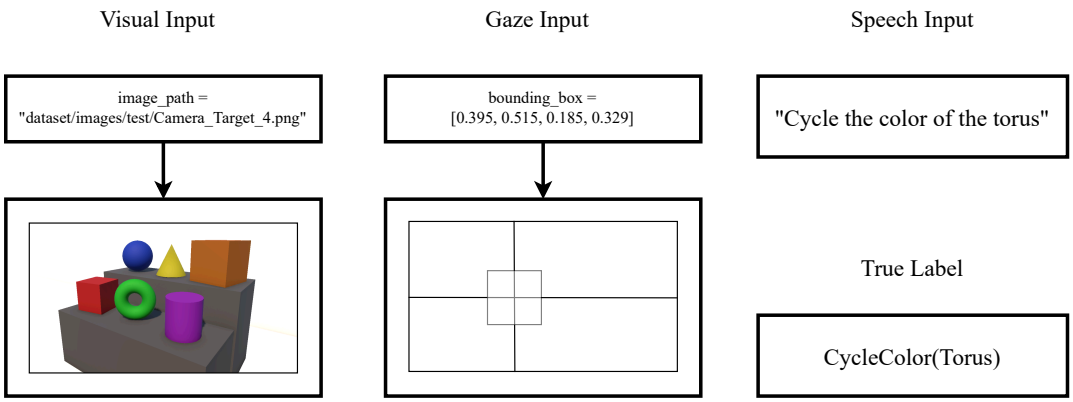


Figure 4.5: Example of an entry in the generated dataset

4.3 Hybrid Fusion Server

To further test the hybrid fusion pipeline, instead of only running it on an isolated script environment, the pipeline was implemented into a local server that could run parallel to the Unity scene, allowing for tests with live data. For this, the speech modality is fully implemented, where a voice-to-text model, Alpha Cephei's "vosk-model-small-en-us-0.15" [3] was used to recognize and process the user's voice commands into the phrases that the textual module was trained for. In addition, as this is not a complete VR scenario but only a simplified representation of the modalities and data involved, the gaze data is provided by the mouse position on the screen when the inference script is run. When activated, this inference script collects the required modal input data and sends it to the server, which runs the pipeline and returns the final user action to the Unity scene.

4.4 Hardware and Software

Regarding the hardware and software used for data generation, model implementation and testing, as well as prototype deployment, all tasks were performed on a single primary workstation to ensure consistency.

- **Workstation Specifications:** 12th Gen Intel(R) Core(TM) i7-12700K CPU, 32 GB of RAM, and an AMD Radeon RX 6950 XT GPU;
- **Data Generation and Testbed:** Unity Engine (Version 6000.0.38f1) with custom C# scripts;
- **Development Environment:** Python 3.12.3.

The model development and deployment were built upon several key open-source libraries.

- **Machine Learning Frameworks:** PyTorch (v2.4.1), Sentence-Transformers (v3.4.1), and Ultralytics (v8.3.82) for model implementation and fine-tuning.
- **Data Handling and Processing:** NumPy (v1.26.4) for numerical operations, Pandas (v2.2.2) for data organization, and OpenCV-Python (v4.9.0.80) for image manipulation tasks.
- **Web Server:** FastAPI was used to build the high-performance API endpoint, which was run on a Uvicorn ASGI server, with Vosk for processing the live audio commands into text.

4.5 Closing Remarks

This chapter details the transition from conceptual design to practical implementation. It includes the creation of a controlled test environment in Unity, a procedurally generated dataset for training and evaluation, as well as fully implementing the two competing pipelines, the modular hybrid

fusion and the specialized late fusion. In addition, it explains the deployment of the hybrid pipeline in a local server in order to validate its potential for real-time applications.

With these fully functional and trained/fine-tuned systems now in place, the next chapter will present the results of this evaluation, comparing performance, resilience, and characteristics of each pipeline, with the aim of answering the research questions posed in Chapter 1.

Chapter 5

Evaluation of the Pipelines

Having designed and implemented the two pipelines in Chapters 3 and 4, this chapter assesses the performance of each when compared to each other, in order to answer the research questions posed in Chapter 1. It aims to discern what challenges may appear when attempting to introduce modularity and VR to an intelligent multimodal framework, as well as how these considerations may affect the performance of said pipelines.

The structure of the chapter is divided into an overview of the experimental setup (section 5.1), the overall results of the pipelines when working as intended (section 5.2), an ablation study of the hybrid pipeline to manage each modality's impact on the model's performance (section 5.2.1), a test including missing data to analyze how the trained pipelines behave if given faulty data (section 5.2.2); and another test to clarify the statistical impact of the textual (speech) data on the model's performance (section 5.2.3). These tests are followed by a discussion and a more thorough review of the research questions (section 5.3).

5.1 Experimental Setup

Both pipelines' MLPs were trained and/or fine-tuned locally for all training and tests, using the generated training dataset for training and validation dataset to evaluate the model after each epoch. For the hybrid fusion pipeline, fully training the MLP took around 3-4 hours. As for the late fusion pipeline, the MLP took < 10 seconds to train fully. The MLPs were trained for 15 epochs in both pipelines with standard cross-entropy loss functions. Specifically, due to the late fusion pipeline's model having two outputs, function and object, respectively, the loss function was instead a joint loss function combining both of the outputs' cross-entropy losses. As for the optimizers, both used PyTorch's Adam optimizer, with a learning rate of 0.001.

As for the other models, fine-tuning the pre-trained Sentence Transformer's "all-MiniLM-L6-v2" model to recognize the target functions took < 1 minute, using a cosine similarity loss function, for 5 epochs, 100 warm-up steps, a learning rate of 2×10^{-5} , and PyTorch's Adam optimizer; It was trained using an image-only dataset, generated similarly to the one used in for the MLP. The image classifier, YOLO's "yolov8s", took around 20 minutes to train for 6 epochs, resizing

images to 640x640 pixels and using PyTorch's Adam optimizer with a learning rate of 0.001 and momentum of 0.9. To properly fine-tune, the model's first 10 layers were frozen, allowing only the head of the model to be trained, using a list of alternatives for each possible command, with variation in the form of synonyms and phrase size and order.

The local server using the hybrid fusion pipeline takes around 25 seconds to start, allows for a 5-second window for a voice command and mouse position to be chosen, and takes around 1 second to return the results to the Unity program.

5.2 Results

Testing the pipelines on the generated testing dataset provides the following results.

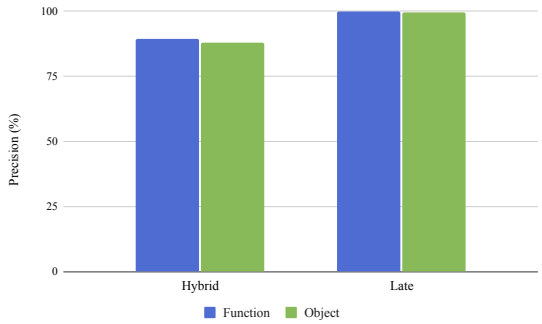


Figure 5.1: Column chart of both hybrid and late fusion pipelines' precisions with all modules

Figure 5.1 shows the precision values for both pipelines, trained and tested using all three modalities against the testing dataset composed of 510 unique entries. It is possible to observe that the late fusion pipeline performs better than the hybrid pipeline, with an average precision of 100% for the given dataset. At the same time, while still effective, the latter only achieves an average of 89.2% for functions and 88% for objects. These results establish a clear performance cost associated with the architectural abstraction required for modularity.

5.2.1 Ablation Study

To test how each modality impacted the results of the hybrid pipeline, an ablation study was performed, where the model was trained N times for the number of modalities, and each time a modality's module was sequentially occluded/removed (e.g., the model was trained without the image module, then tested, then retrained, this time with the image module but without the gaze module, then tested again, etc). In particular, as a consequence of the model's architecture, removing image data from the training and testing datasets inadvertently results in the gaze module not having enough information to be useful.

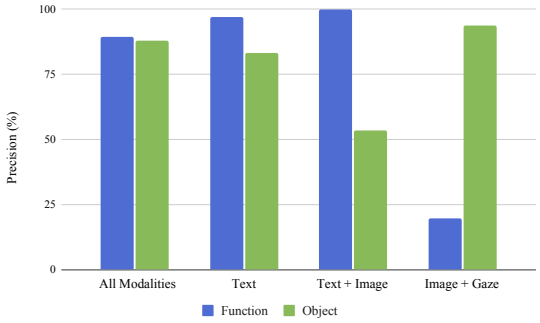


Figure 5.2: Column chart of the ablation study's precisions for the hybrid fusion pipeline

The results in Figure 5.2 show that removing gaze or image data has a negative impact on the performance of the model at deducing the class of the target object while removing textual data entirely removes the model's ability to discern the target function. The lower performance in the 'Text + Image' case, when compared to 'Text', demonstrates that adding un-focused visual data can introduced noise, forcing the model to disambiguate between multiple objects and degrading performance. This study confirms that while all modalities contribute, the hybrid model relies on the synergistic fusion of inputs to achieve its full potential, with text being indispensable for function recognition, specifically.

Notably, this type of analysis was not possible to conduct on the late model, as the model's rigid architecture does not permit it to be functional if one of its specialized inputs is removed. This fact highlights the fundamental trade-off at hand: the late fusion model is robust but inflexible, while the hybrid model is flexible but sensitive to its input configuration.

5.2.2 Missing Data

In order to simulate real-world scenarios where a sensor might fail or an input is missed, both models were tested for faulty inference data. In this test, both models are trained with all modalities, but the data used to test is incomplete.

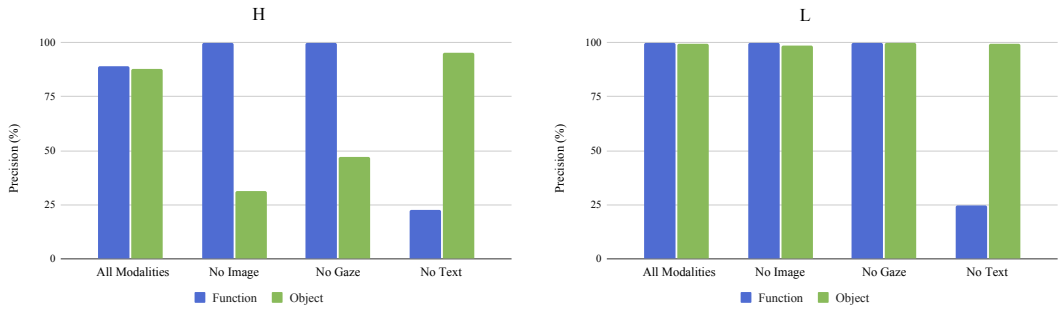


Figure 5.3: Column charts of the precisions when testing data is incomplete, modality-wise; hybrid fusion (H) and late fusion (L)

Figure 5.3 shows that overall, the late fusion pipeline is much more resilient to missing data when compared to the hybrid fusion pipeline. Specifically, the hybrid fusion model becomes unable to discern the target 'Object' if any of the visual or gaze data are missing. The reverse happens with the textual data, the target 'Function' not being discernible if no command is provided. In comparison, the late fusion model is able to reach the correct target 'Object' and 'Function', regardless of image data, and only fails to discern the target function due to the textual data being the only modality providing that context.

This leads to the conclusion that the late fusion pipeline exhibits greater robustness, given its specialized modules. In contrast the hybrid fusion pipeline's performance collapses when either visual or text information are completely absent, demonstrating it is less resilient to the loss of an entire modality.

5.2.3 Textual Impact

In this study, the speech modality, simplified to text for testing purposes, stands out due to its data having both 'Function' and 'Object' information. To test the textual data's impact on both pipelines' performance, the commands given in the testing instances could be:

- **Function + Object:** The full-text command is given (e.g., "Make the cone smaller");
- **Only function:** The text command omits object information, providing only the function (e.g., "Make smaller");
- **Only object:** The text command omits function information, providing only the object (e.g., "Cone");
- **No Data:** The text command contains no data, providing only an empty string (e.g., " ").

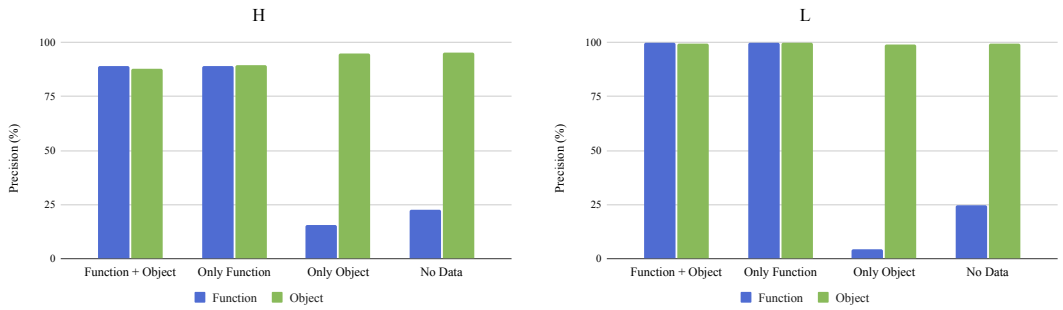


Figure 5.4: Column charts of the precisions when testing the textual data integrity; hybrid fusion (H) and late fusion (L)

As visible in Figure 5.4, both pipelines perform similarly when subject to the test. Removing 'Object' data does not impact the models, as seen in 'Only Function', as both can depend on the other modules to provide the required context to deduce the target object. In contrast, removing

'Function' data in 'Only Object' or 'No Data' results in a sharp decline in the models' ability to deduce the 'Function' information. This behavior is to be expected; as discussed previously, the 'Function' information is only present in the text command; thus, omitting it removes the required context to deduce it correctly.

5.3 Discussion

The reviewed results make it possible to come to some concrete conclusions. Mainly, the late fusion pipeline performed better in almost all the tests than the hybrid counterpart. This outcome, of course, comes at the cost of the desired modularity. In terms of practicality, both setups were quite extensive; however, adapting the hybrid fusion pipeline to different inputs and outputs and other modal combinations becomes much easier compared to the late counterpart. The main drawback of the hybrid fusion approach is the time required to train and fine-tune the models, as it can be substantially slower than in the late fusion pipeline.

This outcome however is not an indictment of the hybrid fusion approach but rather a quantification of its inherent trade-offs. The hybrid model's strength lies in its architectural adaptability, where when compared to the late fusion counterpart, adapting to new modalities becomes a matter of re-training, not re-engineering. The late fusion pipeline, while superior for this specific task, would require significant additional development to accommodate any change. The substantial difference in training time, seconds for the late MLP vs. hours for the hybrid MLP (as shown in section 5.1) further highlights this trade-off between the efficiency of a specialized option and the time-intensive nature of a generalized one.

Additionally, some main problems regarding the hybrid fusion pipeline were not solved. Disambiguation, the ability to discern scene objects from each other, can be essential depending on the context, but this problem was not able to be solved by this research due to its complexity. In a context where the model and pipeline can be fully adapted to the problem at hand, essential syntax like relational positioning (e.g., "the cube in the left"), sizing (e.g., "the bigger cube"), or rotation (e.g., "the cube facing me") become available through a plethora of means. These methods largely depend on more sophisticated MLPs, with specific output channels or larger MLPs, capable of capturing more nuances in the input data. In contrast, these specifications are impossible for the hybrid fusion pipeline, given that the MLP must be abstract and allow for different modal combinations, therefore adapting a "one size fits all" approach.

5.4 Closing Remarks

Through a series of tests, this chapter quantitatively demonstrates the trade-offs between modularity and specialization in intelligent multimodal systems. It established that while a flexible, hybrid architecture is feasible, it currently suffers in both precision and robustness when compared to a specifically purposed late fusion system.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

This dissertation introduced the concept of using AI in the context of multimodal VR environments for more intelligent interactions while maintaining modularity to potentiate the creation of a highly adaptable framework that could be used in any multimodal VR project.

There was an exploration of the state of the art involving currently researched modalities, possible implementations of NNs in multimodal systems, and the intersection of these concepts with VR technologies and applications.

With this exploration, the creation of a hybrid fusion framework that allows for the abstraction required for modularity, with modality-specific modules, whose outputs are combined and processed by an MLP that returns the expected user action, was proposed.

This framework was then implemented, along with a late fusion, non-modular counterpart, to be tested. As an additional requirement and due to restrictions, a custom dataset was also created to train said frameworks.

The analysis performed on the results shows that the late fusion pipeline outperforms the proposed hybrid fusion pipeline by about 10% of average precision, showcasing the upfront cost of modularity. These frameworks were also tested on other parameters, such as resilience to incomplete input data and reliance on the specific textual modality, all presenting a better performance than the late fusion framework. An ablation study was also done to study the impact of each modality on the hybrid fusion framework, showing that each modality positively impacts the pipeline's overall precisions. The same study was not done with the late fusion framework, given that, by lacking modularity, it cannot easily be adapted to other systems, which is the premise of this work.

In conclusion, using AI methods for more intelligent and modular input handling in multimodal VR systems is possible and feasible. However, if the use case allows, the framework developed should be carefully adapted to the problem at hand as the more abstract and modular alternative, due to the emerging and explored restrictions, lacks the specific statistical results to be used practically, especially in VR applications.

6.2 Research Question Analysis

Reiterating, the research questions posed in the section 1.3 can be answered as such:

RQ1: In what ways can AI be used to create a multimodal system for VR environments? As explored in chapter 2, NNs can, among other uses, be implemented as input handling, relieving the programmer of tedious tasks and eventually allowing for more nuanced and complex interactions between the different modalities involved in a project. This approach explored the viability of developing a NN-dependent pipeline for this purpose.

RQ2: What challenges appear in implementing modularity in an intelligent multimodal framework? Chapters 3, 4, and 5 explored the intricacies of implementing modularity within this framework. The designs and implementations of the hybrid and late fusion pipelines, specifically the hybrid counterpart, showcase the restrictions that modularity requires, from a static model to standardized and non-personalized input and outputs. Of note, disambiguation and generalization become relevant problems, where it was found that such an abstract framework cannot, by design, perform with the same care and accuracy as one like the late fusion pipeline, which was explicitly crafted to tackle the case studied. As another objective of the project, the study was explored using only open-source, locally trainable, and access-key-less technologies to allow for not only easy exploration of the matter but also to simulate the system environment of the target user of such a framework.

RQ3: What challenges are involved in applying an intelligent modular multimodal framework to the context of virtual reality? Adding onto the matters of RQ2, the context of VR implies stricter use cases, stricter metrics like response time, precision, faulty data resilience, and a specific set of modalities, most of which have specific applications or require special equipment to be studied. In addition, part of the training process for both pipelines involved the creation of a custom dataset, given the lack of available datasets that involved VR data and models that were complex enough to understand the input data untrained. Due to various constraints, this research explored a use case where only basic and easily accessible modalities were involved.

6.3 Future Work

In the future, the implementation of NNs in this context could be tested in a greater scope, involving a more realistic application, where real users will eventually be tested to compare this approach to more traditional, manual input systems. Focusing more on the work done in this project, the hybrid fusion approach could still be explored by creating a bespoke module manager that avails the target user with easy modality module management, as well as easier NN model training or even trained presets for more general use.

References

[1] Mariano Alcañiz Raya et al. “Machine Learning and Virtual Reality on Body Movements’ Behaviors to Classify Children with Autism Spectrum Disorder”. en. In: *Journal of Clinical Medicine* 9.5 (May 2020). Number: 5 Publisher: Multidisciplinary Digital Publishing Institute, p. 1260. ISSN: 2077-0383. DOI: 10.3390/jcm9051260. URL: https://www.mdpi.com/2077-0383/9/5/1260 (visited on 12/07/2024).

[2] María Alfaro-Contreras et al. “Late multimodal fusion for image and audio music transcription”. In: *Expert Systems with Applications* 216 (2023), p. 119491. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2022.119491. URL: https://www.sciencedirect.com/science/article/pii/S0957417422025106.

[3] Alpha Cephei. *vosk-model-small-en-us-0.15*. 2024. URL: https://alphacephei.com/vosk/models.

[4] Djamila Romaissa Beddiar et al. “Vision-based human activity recognition: a survey”. en. In: *Multimedia Tools and Applications* 79.41 (Nov. 2020), pp. 30509–30555. ISSN: 1573-7721. DOI: 10.1007/s11042-020-09004-3. URL: https://doi.org/10.1007/s11042-020-09004-3 (visited on 12/17/2024).

[5] Joanna Bergström and Kasper Hornbæk. “Human–Computer Interaction on the Skin”. en. In: *ACM Computing Surveys* 52.4 (July 2020), pp. 1–14. ISSN: 0360-0300, 1557-7341. DOI: 10.1145/3332166. URL: https://dl.acm.org/doi/10.1145/3332166 (visited on 12/20/2024).

[6] Mingkai Chen et al. “Cross-Modal Graph Semantic Communication Assisted by Generative AI in the Metaverse for 6G”. en. In: *Research* 7 (Jan. 2024), p. 0342. ISSN: 2639-5274. DOI: 10.34133/research.0342. URL: https://spj.science.org/doi/10.34133/research.0342 (visited on 12/07/2024).

[7] Inrak Choi et al. “Grabity: A Wearable Haptic Interface for Simulating Weight and Grasping in Virtual Reality”. In: *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*. UIST ’17. New York, NY, USA: Association for Computing Machinery, Oct. 2017, pp. 119–130. ISBN: 978-1-4503-4981-9. DOI: 10.1145/3126594.3126599. URL: https://dl.acm.org/doi/10.1145/3126594.3126599 (visited on 12/20/2024).

[8] Chih-Pu Dai and Azibun Nuder. “Extended Reality and Multimodal Artificial Intelligence for Human Performance: A Review of Current Status and Future Outlook”. Preprint available at https://www.preprints.org/manuscript/202410.1023/v1. 2024.

[9] Haolin Fan et al. “Enhancing metal additive manufacturing training with the advanced vision language model: A pathway to immersive augmented reality training for non-experts”. en. In: *Journal of Manufacturing Systems* 75 (Aug. 2024), pp. 257–269. ISSN: 02786125.

DOI: 10.1016/j.jmsy.2024.06.007. URL: https://linkinghub.elsevier.com/retrieve/pii/S0278612524001341 (visited on 12/07/2024).

[10] Massimiliano Gabardi et al. "Development of a miniaturized thermal module designed for integration in a wearable haptic device". In: *2018 IEEE Haptics Symposium (HAPTICS)*. ISSN: 2324-7355. Mar. 2018, pp. 100–105. DOI: 10.1109/HAPTICS.2018.8357160. URL: https://ieeexplore.ieee.org/abstract/document/8357160 (visited on 12/20/2024).

[11] Konrad Gadzicki, Razieh Khamsehashari, and Christoph Zetzsche. "Early vs Late Fusion in Multimodal Convolutional Neural Networks". In: *2020 IEEE 23rd International Conference on Information Fusion (FUSION)*. 2020, pp. 1–6. DOI: 10.23919/FUSION45008.2020.9190246.

[12] Santosh K Gaikwad, Bharti W Gawali, and Pravin Yannawar. "A review on speech recognition technique". In: *International Journal of Computer Applications* 10.3 (2010), pp. 16–24.

[13] Qing Gao, Jinguo Liu, and Zhaojie Ju. "Hand gesture recognition using multimodal data fusion and multiscale parallel convolutional neural network for human–robot interaction". en. In: *Expert Systems* 38.5 (2021). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/exsy.12490, e12490. ISSN: 1468-0394. DOI: 10.1111/exsy.12490. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/exsy.12490 (visited on 10/09/2024).

[14] Google. *Gemini*. 2024. URL: https://gemini.google.com.

[15] Felix G. Hamza-Lup, Kyle Bergeron, and Daniel Newton. "Haptic Systems in User Interfaces: State of the Art Survey". In: *Proceedings of the 2019 ACM Southeast Conference*. ACMSE '19. New York, NY, USA: Association for Computing Machinery, Apr. 2019, pp. 141–148. ISBN: 978-1-4503-6251-1. DOI: 10.1145/3299815.3314445. URL: https://dl.acm.org/doi/10.1145/3299815.3314445 (visited on 12/17/2024).

[16] Takuma Hashimoto et al. "TongueInput: Input Method by Tongue Gestures Using Optical Sensors Embedded in Mouthpiece". In: *2018 57th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*. Sept. 2018, pp. 1219–1224. DOI: 10.23919/SICE.2018.8492690. URL: https://ieeexplore.ieee.org/abstract/document/8492690 (visited on 12/20/2024).

[17] Kaiming He et al. *Deep Residual Learning for Image Recognition*. 2015. arXiv: 1512.03385 [cs.CV]. URL: https://arxiv.org/abs/1512.03385.

[18] Matthew B. Hoy. "Alexa, Siri, Cortana, and More: An Introduction to Voice Assistants". en. In: *Medical Reference Services Quarterly* 37.1 (Jan. 2018), pp. 81–88. ISSN: 0276-3869, 1540-9597. DOI: 10.1080/02763869.2018.1404391. URL: https://www.tandfonline.com/doi/full/10.1080/02763869.2018.1404391 (visited on 12/17/2024).

[19] Juan Izquierdo-Domenech, Jordi Linares-Pellicer, and Isabel Ferri-Molla. "Large Language Models for in Situ Knowledge Documentation and Access With Augmented Reality". en. In: *International Journal of Interactive Multimedia and Artificial Intelligence* In Press.In Press (2023), p. 1. ISSN: 1989-1660. DOI: 10.9781/ijimai.2023.09.002. URL: https://www.ijimai.org/journal/sites/default/files/2023-09/ip2023_09_002.pdf (visited on 12/07/2024).

[20] Katarzyna Jankowiak and Paweł Korpal. “On Modality Effects in Bilingual Emotional Language Processing: Evidence from Galvanic Skin Response”. en. In: *Journal of Psycholinguistic Research* 47.3 (June 2018), pp. 663–677. ISSN: 1573-6555. DOI: 10.1007/s10936-017-9552-5. URL: https://doi.org/10.1007/s10936-017-9552-5 (visited on 12/20/2024).

[21] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. *Ultralytics YOLOv8*. Version 8.0.0. 2023. URL: https://docs.ultralytics.com/pt/models/yolov8/.

[22] Daniel Jurafsky. *Speech and language processing*. 2000.

[23] Erika Kerruish. “Arranging sensations: smell and taste in augmented and virtual reality”. In: *The Senses and Society* 14.1 (Jan. 2019). Publisher: Routledge _eprint: https://doi.org/10.1080/17458927.20 pp. 31–45. ISSN: 1745-8927. DOI: 10.1080/17458927.2018.1556952. URL: https://doi.org/10.1080/17458927.2018.1556952 (visited on 12/20/2024).

[24] May Jorella Lazaro et al. “Multimodal interaction: Input-output modality combinations for identification tasks in augmented reality”. en. In: *Applied Ergonomics* 105 (Nov. 2022), p. 103842. ISSN: 00036870. DOI: 10.1016/j.apergo.2022.103842. URL: https://linkinghub.elsevier.com/retrieve/pii/S000368702200165X (visited on 12/11/2024).

[25] Tsung-Yi Lin et al. *Microsoft COCO: Common Objects in Context*. 2015. arXiv: 1405.0312 [cs.CV].

[26] Sharmin Majumder and Nasser Kehtarnavaz. “Vision and Inertial Sensing Fusion for Human Action Recognition: A Review”. In: *IEEE Sensors Journal* 21.3 (Feb. 2021), pp. 2454–2467. ISSN: 1558-1748. DOI: 10.1109/JSEN.2020.3022326. URL: https://ieeexplore.ieee.org/abstract/document/9187232 (visited on 06/16/2025).

[27] Nader A. Mansour, Ahmed M. R. Fath El-Bab, and Mohamed Abdellatif. “Design of a novel multi-modal tactile display device for biomedical applications”. In: *2012 4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*. ISSN: 2155-1782. June 2012, pp. 183–188. DOI: 10.1109/BioRob.2012.6290860. URL: https://ieeexplore.ieee.org/abstract/document/6290860 (visited on 12/20/2024).

[28] Daniel Martin et al. “Multimodality in VR: A Survey”. In: *ACM Comput. Surv.* 54.10s (Sept. 2022), 216:1–216:36. ISSN: 0360-0300. DOI: 10.1145/3508361. URL: https://doi.org/10.1145/3508361 (visited on 10/30/2024).

[29] Dan Maynes-Aminzade. “Edible Bits: Seamless Interfaces between People, Data and Food”. en. In: (2005).

[30] Microsoft. *Cortana*. 2014. URL: https://www.microsoft.com/en-us/research/group/cortana-research/.

[31] Sushmita Mitra and Tinku Acharya. “Gesture Recognition: A Survey”. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 37.3 (May 2007). Conference Name: IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), pp. 311–324. ISSN: 1558-2442. DOI: 10.1109/TSMCC.2007.893280. URL: https://ieeexplore.ieee.org/abstract/document/4154947 (visited on 12/20/2024).

[32] Jiquan Ngiam et al. “Multimodal deep learning.” In: *ICML*. Vol. 11. 2011, pp. 689–696.

[33] M. A. Nicolaou, H. Gunes, and M. Pantic. "Continuous Prediction of Spontaneous Affect from Multiple Cues and Modalities in Valence-Arousal Space". en. In: *IEEE Transactions on Affective Computing* 2.2 (Apr. 2011), pp. 92–105. ISSN: 1949-3045. DOI: 10.1109/T-AFFC.2011.9. URL: http://ieeexplore.ieee.org/document/5740839/ (visited on 12/18/2024).

[34] Juan Jesus Ojeda-Castelo et al. "A Survey on Intelligent Gesture Recognition Techniques". In: *IEEE Access* 10 (2022). Conference Name: IEEE Access, pp. 87135–87156. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2022.3199358. URL: https://ieeexplore.ieee.org/abstract/document/9858153 (visited on 10/09/2024).

[35] OpenAI. *ChatGPT*. 2024. URL: https://openai.com/chatgpt.

[36] OpenAI. *DALL·E 3 system card*. en-US. Oct. 2023. URL: https://openai.com/index/dall-e-3-system-card/ (visited on 12/17/2024).

[37] OpenAI. *GPT-4V(ision) system card*. en-US. Sept. 2023. URL: https://openai.com/index/gpt-4v-system-card/ (visited on 12/17/2024).

[38] Jieming Pan et al. "A Wireless Multi-Channel Capacitive Sensor System for Efficient Glove-Based Gesture Recognition With AI at the Edge". In: *IEEE Transactions on Circuits and Systems II: Express Briefs* 67.9 (Sept. 2020). Conference Name: IEEE Transactions on Circuits and Systems II: Express Briefs, pp. 1624–1628. ISSN: 1558-3791. DOI: 10.1109/TCSII.2020.3010318. URL: https://ieeexplore.ieee.org/abstract/document/9144270 (visited on 10/09/2024).

[39] G. Pomianos et al. "Recent advances in the automatic recognition of audiovisual speech". en. In: *Proceedings of the IEEE* 91.9 (Sept. 2003), pp. 1306–1326. ISSN: 0018-9219. DOI: 10.1109/JPROC.2003.817150. URL: http://ieeexplore.ieee.org/document/1230212/ (visited on 12/18/2024).

[40] Muhammad Arifur Rahman et al. "Enhancing biofeedback-driven self-guided virtual reality exposure therapy through arousal detection from multimodal data using machine learning". en. In: *Brain Informatics* 10.1 (June 2023), p. 14. ISSN: 2198-4026. DOI: 10.1186/s40708-023-00193-9. URL: https://doi.org/10.1186/s40708-023-00193-9 (visited on 12/07/2024).

[41] Ismo Rakkolainen et al. "State of the Art in Extended Reality - Multimodal Interaction". en. In: (Jan. 2021).

[42] Ananth N. Ramaseri Chandra, Fatima El Jamiy, and Hassan Reza. "A Review on Usability and Performance Evaluation in Virtual Reality Systems". In: *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*. Dec. 2019, pp. 1107–1114. DOI: 10.1109/CSCI49370.2019.00210. URL: https://ieeexplore.ieee.org/abstract/document/9071019 (visited on 06/15/2025).

[43] Nikolaos Rodis et al. "Multimodal Explainable Artificial Intelligence: A Comprehensive Review of Methodological Advances and Future Research Directions". In: *IEEE Access* 12 (2024). Conference Name: IEEE Access, pp. 159794–159820. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2024.3467062. URL: https://ieeexplore.ieee.org/abstract/document/10689601 (visited on 11/25/2024).

[44] Robin Schweigert, Valentin Schwind, and Sven Mayer. "EyePointing: A Gaze-Based Selection Technique". en. In: *Proceedings of Mensch und Computer 2019*. Hamburg Germany: ACM, Sept. 2019, pp. 719–723. ISBN: 978-1-4503-7198-8. DOI: 10.1145/3340764.3344897. URL: https://dl.acm.org/doi/10.1145/3340764.3344897 (visited on 12/17/2024).

[45] Sentence Transformers. *all-MiniLM-L6-v2*. 2025. URL: https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2.

[46] Siri Team. *Hey Siri: An On-device DNN-powered Voice Trigger for Apple's Personal Assistant*. en-US. Oct. 2017. URL: https://machinelearning.apple.com/research/hey-siri (visited on 12/17/2024).

[47] Misha Sra, Xuhai Xu, and Pattie Maes. "BreathVR: Leveraging Breathing as a Directly Controlled Interface for Virtual Reality Games". en. In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. Montreal QC Canada: ACM, Apr. 2018, pp. 1–12. ISBN: 978-1-4503-5620-6. DOI: 10.1145/3173574.3173914. URL: https://dl.acm.org/doi/10.1145/3173574.3173914 (visited on 12/17/2024).

[48] Anirudh Sundar and Larry Heck. *Multimodal Conversational AI: A Survey of Datasets and Approaches*. arXiv:2205.06907. May 2022. DOI: 10.48550/arXiv.2205.06907. URL: http://arxiv.org/abs/2205.06907 (visited on 10/30/2024).

[49] Gemini Team et al. "Gemini: a family of highly capable multimodal models". In: *arXiv preprint arXiv:2312.11805* (2023).

[50] Dangxiao Wang, Kouhei Ohnishi, and Weiliang Xu. "Multimodal Haptic Display for Virtual Reality: A Survey". en. In: *IEEE Transactions on Industrial Electronics* 67.1 (Jan. 2020), pp. 610–623. ISSN: 0278-0046, 1557-9948. DOI: 10.1109/TIE.2019.2920602. URL: https://ieeexplore.ieee.org/document/8733996/ (visited on 12/17/2024).

[51] Zhimin Wang et al. "Interaction With Gaze, Gesture, and Speech in a Flexibly Configurable Augmented Reality System". In: *IEEE Transactions on Human-Machine Systems* 51.5 (Oct. 2021). Conference Name: IEEE Transactions on Human-Machine Systems, pp. 524–534. ISSN: 2168-2305. DOI: 10.1109/THMS.2021.3097973. URL: https://ieeexplore.ieee.org/document/9508143/?arnumber=9508143 (visited on 12/12/2024).

[52] Shengyang Xu et al. "LLM enabled generative collaborative design in a mixed reality environment". en. In: *Journal of Manufacturing Systems* 74 (June 2024), pp. 703–715. ISSN: 02786125. DOI: 10.1016/j.jmsy.2024.04.030. URL: https://linkinghub.elsevier.com/retrieve/pii/S0278612524000967 (visited on 12/07/2024).

Appendix A

Test Data

Table A.1: Precisions (0-1) used in basic fusion comparison

Category	Item	Hybrid	Late	N. of Entries
Function	ChangeMaterial	0.94	1	90
	CycleColor	0.76	1	98
	Destroy	0.89	1	108
	MakeBigger	0.87	1	100
	MakeSmaller	1	1	114
	Avg :	**0.892**	**1**	**510**
Object	Cone	0.95	0.98	101
	Cube	1	1	101
	Cylinder	0.72	1	103
	Sphere	0.83	1	104
	Torus	0.9	1	101
	Avg :	**0.88**	**0.996**	**510**

Table A.2: Precisions (0-1) used in hybrid fusion pipeline's ablation study

Category	Item	All Modules	Text	Text + Image	Image + Gaze
Function	ChangeMaterial	0.94	0.85	1	0.23
	CycleColor	0.76	1	1	0
	Destroy	0.89	1	1	0.2
	MakeBigger	0.87	1	1	0.28
	MakeSmaller	1	1	1	0.27
	Avg :	**0.892**	**0.97**	**1**	**0.196**
Object	Cone	0.95	0.78	0.49	0.9
	Cube	1	0.58	0.78	0.97
	Cylinder	0.72	1	0.52	0.92
	Sphere	0.83	0.79	0.44	0.9
	Torus	0.9	1	0.44	0.99
	Avg :	**0.88**	**0.83**	**0.534**	**0.936**

Table A.3: Precisions (0-1) used in missing data study

Category	**Item**	**Hybrid Fusion**				**Late Fusion**			
		All Modules	No Image	No Gaze	No Text	All Modules	No Image	No Gaze	No Text
Function	ChangeMaterial	0.94	1	1	0.35	1	1	1	0.4
	CycleColor	0.76	1	1	0.19	1	1	1	0.15
	Destroy	0.89	1	1	0.12	1	1	1	0.26
	MakeBigger	0.87	1	1	0.19	1	1	1	0.22
	MakeSmaller	1	1	1	0.27	1	1	1	0.2
	Avg :	**0.892**	**1**	**1**	**0.224**	**1**	**1**	**1**	**0.246**
Object	Cone	0.95	1	0.58	0.97	0.98	0.94	0.99	0.98
	Cube	1	0.3	0.84	1	1	1	1	1
	Cylinder	0.72	0.26	0.23	0.93	1	1	1	1
	Sphere	0.83	0	0.71	0.91	1	1	1	1
	Torus	0.9	0	0	0.95	1	1	1	1
	Avg :	**0.88**	**0.312**	**0.472**	**0.952**	**0.996**	**0.988**	**0.998**	**0.996**

Table A.4: Precisions (0-1) used in text data comparison

Category	**Item**	**Hybrid Fusion**				**Late Fusion**			
		Full Text	Only Function	Only Object	No Data	Full Text	Only Function	Only Object	No Data
Function	ChangeMaterial	0.94	0.92	0.33	0.35	1	1	0	0.4
	CycleColor	0.76	0.86	0	0.19	1	1	0	0.15
	Destroy	0.89	0.82	0.26	0.12	1	1	0.21	0.26
	MakeBigger	0.87	0.85	0.18	0.19	1	1	0	0.22
	MakeSmaller	1	1	0	0.27	1	1	0	0.2
	Avg :	**0.892**	**0.89**	**0.154**	**0.224**	**1**	**1**	**0.042**	**0.246**
Object	Cone	0.95	0.95	1	0.97	0.98	0.99	0.97	0.98
	Cube	1	0.98	1	1	1	1	1	1
	Cylinder	0.72	0.73	0.89	0.93	1	1	1	1
	Sphere	0.83	0.87	0.95	0.91	1	1	0.99	1
	Torus	0.9	0.94	0.91	0.95	1	1	0.99	1
	Avg :	**0.88**	**0.894**	**0.95**	**0.952**	**0.996**	**0.998**	**0.99**	**0.996**